

GIBSON DUNN

Gibson, Dunn & Crutcher LLP

200 Park Avenue
New York, NY 10166-0193
Tel 212.351.4000
www.gibsondunn.com

Daniel J. Thomasch
Direct: +1 212.351.3800
Fax: +1 212.351.6200
DThomasch@gibsondunn.com

October 28, 2011

The Honorable Robert E. Payne
United States District Court
Eastern District of Virginia
Spottswood W. Robinson III and
Robert R. Merhige, Jr., Federal Courthouse
701 East Broad Street
Richmond, Virginia 23219

Re: ePlus, Inc. v. Lawson Software, Inc., Civil Action No. 3:09CV620

Dear Judge Payne:

As counsel for Defendant Lawson Software, Inc. in connection with the allegations of contempt, we have received the Court's request for a telephonic hearing this afternoon, as well as for an in-person conference on November 3, 2011. We will, of course, be available for both conferences, and in advance of the in-person conference will endeavor to provide any additional information or briefing that may assist the Court.

Prior to these conferences, I would like to briefly respond to three issues contained in Mr. Merritt's October 26, 2011 letter to the Court on behalf of ePlus: 1) ePlus's characterization of the nature and content of the upcoming proceedings, 2) the appropriate scope of discovery in contempt proceedings, and 3) the schedule for these proceedings.

To begin with, ePlus's statement of the issue to be adjudicated at the contempt proceeding cannot be harmonized with controlling authority. According to ePlus, the sole issue at the upcoming proceeding is "the viability of Lawson's assertion that the" newly accused RQC product is "non-infringing." But in *Tivo Inc. v. EchoStar Corp.*, 646 F.3d 869 (Fed. Cir. 2011), the Federal Circuit made clear that the issue of infringement is not the primary one in these proceedings. Instead, the party seeking to enforce the injunction must first prove that the newly accused product is not more than colorably different from the product found to infringe. *Id.* at 882. The colorable differences test is *different* from the question of whether the newly accused product infringes the patent claims at issue and those separate analyses may not be conflated. *Id.* After all, if the differences are more than colorable, the defendant is not in contempt and the proceedings must end *before* infringement is analyzed. If (and only after) the party seeking to enforce the injunction proves that the differences are merely colorable, it then must prove that the newly accused product infringes the asserted claims. *Id.* at 883. In enunciating this new framework, the Federal Circuit flatly rejected any notion that an alleged contemnor must show cause as to why it is not in contempt, as ePlus seeks: "The patentee bears the burden of proving violation of the injunction by clear and convincing evidence, a burden that applies to both infringement and colorable differences." *Id.*

GIBSON DUNN

The Honorable Robert E. Payne
October 28, 2011
Page 2

The framework set forth in the *Tivo* decision is directly relevant to the second issue raised in ePlus's letter: the proper scope of discovery in contempt proceedings. Even before its recent production of nearly 1 million pages of documents, Lawson already had provided ePlus with comprehensive discovery that demonstrates the more than colorable differences between the adjudged infringing RSS product and the redesigned RQC product.¹ Among other things, Lawson provided ePlus with the source code to the RQC product, a series of source code difference files that clearly show the source code changes made to the RSS product, as well as a fully-loaded laptop that permits ePlus's expert witness, Dr. Weaver, to undertake any demonstration of the RQC product that he sees fit. Moreover, at ePlus's request, and without objection, Lawson has provided interrogatory responses to ePlus, including a 32-page Supplemental Answer to Lawson's Response to Interrogatory No. 5, which sets forth both in a plain language summary and in extensive detail with citations to underlying code changes the more than colorable differences between the RSS and RQC product configurations, which also render the RQC configuration non-infringing. A copy of that Supplemental Answer is attached hereto for the Court's review.

In addition to the foregoing, Lawson will be providing corporate representatives to testify to 22 separate subject matters contained in a Rule 30(b)(6) deposition notice served by counsel for ePlus. Here again, Lawson did so without objections despite the fact that the number of subject matters far exceed what ePlus had represented to the Court it would need. Sept. 15 Tr. at 9 (Mr. Robertson: "So, you know, the 30(b)(6) deposition shouldn't be limited to five topics. Our topics are about ten or 12.").

In short, ePlus now has in its possession everything it needs (and more) to determine whether the changes were more than colorable, and whether the newly accused product infringes.

If, in the face of ample evidence documenting the significant changes Lawson has made to its product, ePlus nevertheless wishes to proceed with its allegations of contempt, procedural fairness – as well as controlling authority – demands that Lawson be informed with particularity of the basis for the serious charges against it. *See Tivo*, 646 F.3d at 881 ("What is required for a district court to hold a contempt proceeding is a detailed accusation from the injured party setting forth the alleged facts constituting the contempt.").

¹ The more than 1 million pages of documents responsive to ePlus's very broad document production requests and search terms were *not* needed to show the precise changes at issue. The extraordinary scope of this production, as well as the timing of the production, is directly attributable to the breadth of ePlus's document requests and search terms. On October 7, 2011, nearly three weeks after ePlus had served its discovery requests, ePlus identified for the first time search terms that it demanded Lawson use for its document collection, review, and then production. Those search terms – which included words such as "*valid*", "injunct*", and "jury" – were guaranteed to capture many more documents than those necessary to compare the features of the newly accused RQC product to the adjudged infringing RSS product.

GIBSON DUNN

The Honorable Robert E. Payne
October 28, 2011
Page 3

Moreover, Lawson is entitled to take discovery from ePlus to determine what analysis ePlus has performed to compare the newly accused product and the adjudged infringing product, as well as to determine what statements ePlus has made to third parties (including current and potential customers) about the same issue. That Lawson possesses information (which it has produced to ePlus) showing differences between the adjudged infringing product and the newly accused product does not mean that no other information supports its defense. For instance, Lawson should be entitled to bring to the Court's attention admissions made by ePlus employees as to differences between those two products, if such evidence exists. Lawson also is entitled to determine the remedies ePlus seeks at these proceedings, and the alleged bases for those remedies.

The discovery Lawson has served on ePlus is narrowly tailored to these goals and generally can be grouped into one of three categories:

1. ePlus's contentions in these proceedings, as well as the facts relevant those contentions.
2. Analysis performed by or on behalf of ePlus to compare the newly accused RQC product to the adjudged infringing RSS product.
3. The remedies ePlus seeks in these proceedings (which would include any impact that the newly accused RQC product has had on ePlus's allegedly competing products).

The intent of these discovery requests is not, as ePlus would have it, "re-litigation." The RQC product is a new design that was not at issue in the earlier trial, so re-litigation would have no purpose. Because the issues to be adjudicated in the contempt proceedings are new – and because different facts would be needed to prove ePlus's new allegations – discovery obligations should be reciprocal. Lawson has made its disclosures and is now entitled to understand the evidence that ePlus possesses relevant to the new issues to be tried.

In its October 26, 2011 letter, ePlus also raised the prospect of delaying the scheduled evidentiary proceedings. Lawson is not opposed to such a delay (or a delay in ePlus's responses to Lawson's discovery requests), provided that any new schedule is implemented in a manner consistent with the parties' disclosure obligations and other pretrial considerations. In that regard, Lawson believes that the Court would benefit from focused briefing that identifies the specific matters to be resolved at the proceedings as well as advance disclosure by both parties of the witnesses (including expert witnesses) and exhibits to be used at the proceedings.

GIBSON DUNN

The Honorable Robert E. Payne
October 28, 2011
Page 4

Lawson will be prepared to address these issues, as well as any other issues the Court may wish to raise, at the November 3, 2011 conference.

Sincerely,


Daniel J. Thomasch

DJT/las

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF VIRGINIA
Richmond Division

ePLUS INC.,)
)
) Civil Action No. 3:09cv620 (REP)
Plaintiff,)
)
v.)
)
LAWSON SOFTWARE, INC.,)
)
)
Defendant.)

DEFENDANT LAWSON SOFTWARE, INC.'S SUPPLEMENTAL ANSWERS TO
PLAINTIFF ePLUS INC.'S FIRST SET OF INTERROGATORIES
FOR CONTEMPT PROCEEDINGS

Pursuant to Rule 33 of the Federal Rules of Civil Procedure, Defendant Lawson Software, Inc., ("Lawson") hereby serves the following Answers to Plaintiff ePlus, Inc.'s ("ePlus") First Set of Interrogatories for Contempt Proceedings.

INTERROGATORIES

Interrogatory No. 5:

Identify with specificity any portions of the Core S3 Procurement System source code that were modified during the development and creation of RQC.

ANSWER:

Lawson objects to the interrogatory to the extent it seeks information protected by the attorney-client privilege and the common interest privilege. Subject to and without waiving the foregoing objections, Lawson states: Pursuant to Fed. R. Civ. P. 33(d), each of the source code changes to Lawson's Core S3 Procurement System is identified in the document labeled RQC0000748, which was produced on August 17, 2011.

SUPPLEMENTAL ANSWER:

Lawson's Supplemental Answer will address two aspects of Lawson's product: Lawson's Core S3 Procurement System (which as a standalone system was not accused of infringing, or adjudged to infringe, the patents at issue), as well as Lawson's newly-designed RQC product (which is a software module that is separate from, but works in conjunction with, a modified version of the Core S3 Procurement System).

A. Overview of Lawson's Supplemental Answer to Interrogatory No. 5

To summarize, Lawson modified the source code for the Core S3 Procurement System in order to prohibit users from punching out to more than one vendor. In the adjudged infringing systems that incorporated RSS, Punchout, and the Core S3 Procurement System, users were able to punch out to two or more vendors (e.g., Dell and Staples) in the same requisition. Changes made to the source code in connection with the Core S3 Procurement System now prevent users of the RQC module from having the ability to punchout to two or more vendors in the same

requisition. Now, items from one punchout vendor (*e.g.*, Dell) must be purchased or deleted from a requisition before a user is permitted to view – much less select – items from a second punchout vendor (*e.g.*, Staples). A warning message has been added to inform users who attempt to combine items from two or more punchout vendors that such combinations are not permitted.

Lawson also modified the adjudged infringing RSS module in significant ways to create the newly-accused RQC software module. In summary, the changes detailed below include the following:

First, the RQC product no longer includes a shopping cart that is distinct from the requisition. In the RSS product, users would place selected items into a shopping cart, which ePlus accused of constituting an “order list.” When the user initiated a “Save” or “Checkout” operation, the software utilized the contents of that cart to build a new requisition, which ePlus accused of constituting a “requisition . . . relating to selected matching items on said order list.” The newly-accused RQC product does not contain an intermediate shopping cart; instead, user-selected items now are saved directly to the requisition. This is an actual substantive change in the manner in which the system operates, and not a mere matter of nomenclature as has been suggested by ePlus. What the user sees on the screen as a “Requisition Line” is not, as ePlus contends, an intermediate list which is then used to build a requisition. Instead, what the user sees as a “Requisition Line” accurately reflects that which *already* has been stored in the underlying requisition database, which was not the case in the adjudged infringing system.

Second, the RQC product eliminates a user’s ability to search for products by the fourth, most specific, level (*i.e.*, the seventh and eighth digits) of the UNSPSC code. At trial, ePlus alleged that this fourth level searching satisfied the “converting data relating to a selected

“matching item” claim limitations present in claims 3, 38, and 29 of the ’683 patent because items grouped with the same commodity level code share common basic characteristics. The RQC product eliminates altogether a user’s ability to search or to group products by the fourth level of the UNSPSC code.

Third, the RQC product does not provide the capability to combine items from Item Master and a punchout vendor in a single requisition. This is, on its face, more than a merely colorable change to the operation of the product at issue. Moreover, by eliminating the capacity of the adjudged infringing product to combine items from Item Master and a punchout vendor in a single requisition, the newly-accused product cannot be found to infringement any of the asserted claims of the ’683 patent, every one of which requires the ability to combine items from “at least two product catalogs.” (While counsel for ePlus argued at trial that Item Master alone constituted “at least two product catalogs,” without regard to Punchout, the jury rejected that theory of infringement in finding Configuration Nos. 2 and 4 to be not infringing). Users of RQC are limited to requisitions from only a single source. Once a requisition includes items from Item Master, the RQC software prohibits the user from initiating a punchout session – much less selecting for purchase items from a punchout vendor. Similarly, once a requisition includes items from any punchout vendor, the RQC software prohibits the user from searching for – much less selecting for purchase – items from the Item Master database or a second punchout vendor.

The changes to the noninfringing Core S3 Procurement System are addressed because as drafted, Interrogatory No. 5 requests that Lawson describe only changes to the Core S3 Procurement System, which itself was not held to infringe any asserted claim of the patents-in-suit. Indeed, ePlus did not even accuse certain Lawson software configurations that included the

Core S3 Procurement System of infringing the claims at issue. Moreover, ePlus accused of infringement two other Lawson software configurations that included the Core S3 Procurement System (Configuration Nos. 2 and 4); but the jury determined that both configurations did not infringe the patents at issue.

To be clear, the Core S3 Procurement System is different and separate from the former RSS module, previously judged infringing, as well as the new RQC module. The source code that relates to the Core S3 Procurement System was separate from the RSS source code, and remains separate from the RQC source code. Nonetheless, because the Core S3 Procurement System underlies the new RQC product, Lawson does not object to describing changes to the noninfringing Core S3 Procurement System. Below, Lawson details three substantial source code modifications that were implemented to the noninfringing Core S3 Procurement System.

The changes to the newly-designed RQC product are addressed because ePlus's October 11, 2011 letter requested information that is different from – and broader in scope than – the information requested in the Interrogatory. Specifically, ePlus requested "an identification of the portions of the source code of the infringing system configuration that were 'modified during the development and creation of RQC.'" This late request apparently includes not only changes to the source code for the noninfringing Core S3 Procurement System, but also changes to the RSS source code and the creation of the re-designed RQC source code.

Despite the fact that Interrogatory No. 5 does not call for a summary of the differences made to the RSS system to produce the RQC system, Lawson responds in full to provide a detailed summary of the modules that were changed in the RSS and Punchout configurations. Below, Lawson details three sets of changes to the infringing RSS system that makes RQC more

than colorably different from RSS, thereby avoiding infringement of all of the claims the jury found to be infringed.

We address first the changes to the non-infringing Core S3 Procurement System, and second, the changes made to the infringing RSS module.

B. Changes Made To The Noninfringing Core S3 Procurement System

The changes Lawson made to the Core S3 Procurement System all relate to restricting the ability for a user to punch out to more than a single vendor during a single requisition session. Three groups of changes were made to the source Core S3 Procurement System source code to effectuate the restriction. First, the source code has been modified to store a value of the Punchout vendor in the core database. Second, the RQC system prohibits users from punching out to multiple vendors. Finally, the RQC system changed the field label messages and brings the language used in the new RQC program into the core system.

First, the source code was modified to store a value of the Punchout vendor in the core database. This change restricted a user from exiting out of a non-final requisition, later coming back to that requisition, punching out to a second vendor, and placing items from that second vendor on the first requisition. Lawson stores the value in an existing database field that is not used when doing a Punchout Requisition. A database change was done to create a proper holding area. The source code change to effectuate this result is illustrated on at least the following pages of source code as set forth in RQC0000748:

- **Page 298 (note related to JT-250022 Allow only one punchout vendor item per requisition)**
 - o Page 342 (JT-250022 AND (RQRQH-RLN-COST-CD NOT = "G"))
 - o Page 386 (JT-250022 AND (RQRQH-RLN-COST-CD NOT = "G"))
 - o Page 399 (JT-250022 AND (RQRQH-RLN-COST-CD NOT = "G"))
 - o Page 400 (JT-250022 AND (RQRQH-RLN-COST-CD NOT = "G"))
 - o Page 401 (JT-250022 AND (RLN-COST-CD NOT = "G"))
 - o Page 403 (JT-250022 AND (RLN-COST-CD NOT = "G"))
 - o Page 437 (JT-250022 AND (RQRQH-RLN-COST-CD NOT = "G"))
 - o Page 456 (JT250022 AND (PLI-COST-CD NOT = "G"))

- o Page 479 (JT250022 AND (PLI-COST-CD NOT = "G"))
- o Page 481 (JT250022 AND (PLI-COST-CD NOT = "G"))
- o Page 482 (JT250022 AND (PLI-COST-CD NOT = "G"))
- o Page 496 (JT250022 AND (POPCR-PLI-COST-CD NOT = "G"))
- o Page 510 (JT250022 AND (POPIF-PIF-COST-CD NOT = "G"))
- o Page 577 (JT-250022 AND (POPCR-PLI-COST-CD NOT = "G"))
- o Page 698 (JT-250022 AND (RLN-COST-CD NOT = "G"))
- o Page 698 (JT-250022 AND (PLI-COST-CD NOT = "G"))
- o Page 708 (JT-250022 AND (RLN-COST-CD NOT = "G"))

Second, the Core S3 Procurement System was modified so that the RQC system prohibits users from punching out to multiple vendors. The messaging to the user informed the user that the user cannot select items from a second punchout vendor. The user may either cancel out of going to a new vendor or create a new requisition with the new vendor. If the user chooses to move forward with a new requisition, a new requisition is created, and the header information from the originating requisition without item information is preserved on the new requisition. The source code change preserved the requisition header information without preserving the item information. The source code related to this change includes the following set forth below:

- **Page 279 (note related JT-245782 – add option to RQIJ to skip the copying of the requisition lines during the copy requisition option)**
 - o Page 284 IF (RQIJF1-SKIP-LINES = "Y") (additional code in RQC0000748)
 - o Page 296 (| Skip Lines: [])
 - o Page 297 (| 3 SKIP-LINES A 01 SB)

Third, the Core S3 Procurement System was modified so that the RQC system changed the field label messages and brings the language used in the new RQC program into the core system. These code changes occurred at least in the following locations:

- **Page 2 (note related to J-233000 Changes to coincide with Requisition Center Changes)**
 - o Page 50 at 170789 (Move "Unreleased" To POXMLWS-NODE-VALUE)
 - o Page 109 at NODE (Move "Template" To POXMLWS-NODE-VALUE)
 - o Page 191 (Move "Unreleased" To POXMLWS-NODE-VALUE)
 - o Page 204 (Move "Unreleased" To POXMLWS-NODE-VALUE)
 - o Page 252 (\$TAB-FORM EREQS "Requisition Center")
 - o Page 253 (| Template:& | Punchout:& |)
 - o Page 259 (\$TAB-FORM E-REQS "Requisition Center")

- o Page 260 (| Template: &] | Punchout: &])

The full context of the code changes is found in RQC0000748.

On August 12 and August 17, 2011, Lawson voluntarily and pro-actively provided source code difference files to ePlus regarding the Core S3 Procurement System, RSS, and RQC products. Lawson provided those documents to ePlus prior to the current proceedings in order to clarify that the newly-accused products are substantially different from the adjudged infringing products, and do not infringe any of the claims the jury found to be infringed by Lawson. The document labeled RQC0000748 specifies the changes that were made to the Core S3 Procurement System source code. The source code differences for each module relating to the RSS and Punchout products are set forth in detail in source code differences files that were produced as RQC0000746, RQC0000749, and RQC0000750.¹

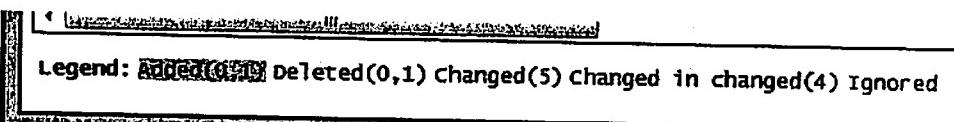
¹ The source code differences files were created using the ExamDiff Pro, a prestoSoft software application. More information related to that software program can be found at http://www.prestosoft.com/edp_examdiffpro.asp.

C. Changes Made To The RSS Module To Obtain The Noninfringing RQC Product

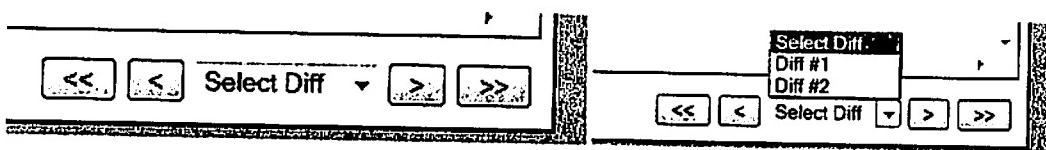
Source code difference files were created to show the source code differences between RSS and RQC. That software displays the RSS code on the left side of a split screen and the RQC code on the right side of a split screen. A depiction of the source code differences is set forth below:

The screenshot shows a split-screen interface of the ExamDiff Pro software. The left pane displays the RSS source code, and the right pane displays the RQC source code. The code is color-coded to highlight differences: red for deleted text, green for added text, and blue for changed text. A legend at the bottom left of the interface provides a key for these colors.

The source code changes are color coded. A legend at the bottom left of the ExamDiff Pro screen shows the color codes. Those codes are blown up in the image below:



A user of the ExamDiff Pro software may go through the source code by using the scroll bar on the right side of either the RSS or RQC source code. Alternatively, the user may use select the difference by using the jump menu at the bottom right of the software, which is depicted below in both the static state and showing the difference numbers:



Each module or file that Lawson modified was produced in RQC0000746, RQC0000749, and RQC0000750. Each file details the changes that were made to the source code. Specific portions of the difference files are referenced in this response. However, because the changes must be viewed in context with the overall source code, Lawson incorporates those documents in this response.

The following chart summarizes the modifications to the source code and each source code module that was modified to effectuate the change. The header of the table describes the modification at issue. The left column of the table details each source code module that was modified to effectuate the modification.

For example, to determine the source code files that were altered to restrict requisitions from including items from the Item Master and a Punchout vendor, one would look at the first column (from the left) of the table. If a check mark is present in a specific row, that check mark reflects that the particular source code file was modified to effectuate this change. In the case of the restriction on requisitions including items from the Item Master and a Punchout vendor, the following files were modified: Cart.js, Dialog.js, inquire.js, requisition.js, and shoppingrtns.js.

File Name	Separate punchout and Internally sourced items	Label / Visual / Messages...	Punchout Disclaimer	One punchout vendor per req	Immediately add / change / remove data from and to DB	Only show 3 UNSPSC levels	Remove branding frame when punching out
application.xml			✓				
approvalListReqs.htm			✓				
Cart.js	✓					✓	
CartList.java					✓		
commonrtns.js		✓					
debug.htm		✓					
debug.js		✓					
debugApprovals.htm		✓					✓
debugRec.htm		✓					
debugReqStatus.htm		✓					
debugTemplates.htm		✓					

default.xml		✓				
Dialog.js	✓	✓		✓		✓
ico_cart_green.gif(Deleted)		✓				
ico_cart_trans.gif(Deleted)		✓				
index.htm		✓				
index.htm(2)		✓			✓	
indexLocal.htm		✓				
inquire.js	✓					
inquireLine.htm		✓				
inquireOrder.htm		✓				
LAWSONLD.shopping.csv		✓				
LineItem.js					✓	
logan.rqc.bkmk(Added)		✓				
logan.rqc.bkmk(Deleted)		✓				
menus.js		✓				
messageDialog.htm		✓				
messageDialog.js		✓				
messageDialog.js(2)		✓				
pdf.htm		✓				
punchout.htm			✓			
punchout.js			✓	✓	✓	✓
PunchoutOrderMessage.java				✓		✓
punchoutShop.htm						✓
punchoutShop.js						✓
PunchoutStepsActionHandler.java		✓				
reqCartDetail.js				✓		
reqCartDetail.xsl				✓		
reqLineStatusList.htm		✓				
reqStatusList.htm		✓				
Requisition.js	✓	✓		✓	✓	
rqc_config.xml(Added)		✓				
rss_config.xml(Deleted)		✓				
RQParser.java				✓		
rss.js					✓	
RSSShell.js		✓				
shoppingrtns.js	✓					✓
StylerBase.js		✓				
StylerRSS.js		✓				
Template.js		✓				
templateCopy.htm		✓				
templateNew.htm		✓				
templatePrint.htm		✓				
utility.js		✓				

✓
web.xml
XRQLineDetails.java
✓
4GL changes
✓

Lawson set forth the description of the source code changes in response to Interrogatory No. 1 and incorporates that portion of its Response to Interrogatory No. 1 herein. A further summary of the changes was set forth at pages 1-2 of Lawson's opposition to ePlus's motion to show cause (Dkt. No. 804), which Lawson incorporates herein. The source code changes to each of the modifications are set forth below:

1. **Lawson removed the shopping cart functionality and required that any product selected under the Find/Shop results would go directly to the requisition, therefore bypassing the intermediate step of a shopping cart**

Lawson implemented a source code change that directed any item selected from a Find/Shop task directly to a requisition, which was the functionality employed by the prior art, including Lawson's own RQ module. This redesign precludes infringement of Claim 1 of the '172 Patent. The source code modules modified to implement this change are Cart.js, debug.js, default.xml, index.htm(2), LineItem.js, punchout.js, Requisition.js, and rss.js. The differences between the RSS and RQC versions of those files are shown in RQC0000750.

Claim 1 of the '172 Patent requires an "order list." ePlus contended at trial that the order list for Configurations 2, 3, and 5 was the RSS "shopping cart", which was labeled "My Cart". With RSS, no requisition entries were made from the My Cart (what ePlus argued to be the order list) to RQ10 (the requisition) until after items were selected to populate a requisition, when the user clicked the "SAVE" or CHECKOUT buttons. The RSS systems separated out the My Cart and the Requisition as two separate steps. ePlus contended at trial through the testimony of Dr. Weaver that the RSS shopping cart is different than a requisition. He stated "the order list is the shopping cart and that's what becomes the requisition." (Trial Trans. at 569:8-9.) In response to

the question “Is [a shopping cart] consistent with your understanding of building a requisition,” Dr. Weaver testified, “Well, it’s not the requisition... [the order list] is going to be transferred to the requisition module.” (Trial Tr. at 568:22-569:5); *see also* (Trial Trans. at 568:14-19.) Figure 1C of the ‘172 patent confirms that the requisition is different from the order list. Dr. Weaver’s expert report provides further support for the difference between the order list and the requisition. Weaver Expert Report on Infringement, dated 5/5/2010 at p.85. Additionally, ePlus admitted that separate order lists and requisitions are required to overcome prior art. “Rather, the P.O. WRITER system can start with building a requisition or building a purchase order (without even building a requisition). If the user starts with a requisition, from the search results generated in that process, the system builds a requisition, not an order list.” Hilliard Expert Report, dated 6/9/2010, at p. 58 (regarding ‘172 claim 1). “Nor does the system disclosed in the King Patent perform the intermediate step of generating an order list that includes at least one matching item selected by said means for searching. Any search results obtained from the searches in the Public Catalog or Private Catalog are entered directly onto the requisition.” Hilliard Expert Report, dated 6/9/2010, at p. 115 (regarding ‘172 claim 1). Further reference to ePlus’s admissions that the order list and requion must be different and that the order list is an intermediate step needed prior to the generation of the requisition is set forth in September 19, 2011, declaration of Dr. Michael Shamos at Paragraphs 20-23 and documents cited therein.²

Based on ePlus’s admissions and representations at trial, and admissions related to existing prior art, Lawson removed the intermediate step of sending selected items to an “order list,” and instead modified the source code to deliver the selected items directly to the requisition. The modified RQC product operates like the prior art and the non-infringing RQ10

² The attached admissions by ePlus are included as context for the materiality of the changes described in this Supplemental Response. They are merely examples, and not intended to be comprehensive listings of all evidence showing ePlus’s positions at trial.

module. In turn, the user interface called “Requisition Lines” (visually in the same location as the former My Cart) simply displays the contents of the requisition after the requisition has been updated. It cannot display the contents of a separate order list because there is no separate order list.

A detailed analysis of the comparisons between RSS and RQC related to the removal of the “means for generating an order list” is set forth in paragraphs 15-40 of the Declaration of Defendant’s Expert Michael I. Shamos, Ph.D, J.D. Concerning Lawson’s Non-Infringing Design Around dated September 19, 2011, which is incorporated by reference herein. Further, Lawson incorporates by reference pages 7-15 of Dkt. No. 804, which relates to the removal of the order list.

The revised manner in which data flows demonstrates the significance of this change. In RSS, the data flowed from the search results to the My Cart to the requisition (with a one-time error check of all of the items at the end. With RSS, the user must initiate an extra mandatory step to move an item from the order list to a requisition in RQ10.1. The user moves an item from the order list to the requisition by clicking “SAVE” button or the “CHECKOUT” button. The source code showing that option in RSS is set forth below. This functionality was commented out in the RQC system because the items no longer reside in a temporary My Cart, but are sent directly to the requisition:

Index.htm

```
420 <!-- EP td><button style="pushbutton" id="saveCartBtn" class="button"
onclick="requisition.save()" disabled></button></td><!-- Save -->
```

Cart.js

```
36 // EP this.saveBtn = document.getElementById("saveCartBtn");
```

```
321 // var btnAry = new Array(this.newBtn, this.clearBtn, this.printBtn, this.saveBtn, this.checkoutBtn);  
951 // EP this.saveBtn = document.getElementById("saveCartBtn");  
958 // EP this.saveBtn.disabled = true;  
977 // EP  
978 // if (sipObj.currentPage == PROFILE_PAGEID)  
979 //     this.saveBtn.disabled = (this.isSavedToDB() && !mainWnd.WORK.dataHasChanged) ? true : false;  
980 // else  
981 //     this.saveBtn.disabled = this.isSavedToDB();  
982 // EP  
990 // EP mainWnd.sipObj.styler.processButtonElement(window, this.saveBtn); // restyle button
```

rss.js

```
44 // EP translate("saveCartBtn", "Save");
```

Upon Checkout, the RSS software checks all of the items for errors.

In RQC, by contrast, the data flows from the search results directly to the requisition (whereupon the addition of the item is error checked immediately on an item-by-item basis prior to requisition), skipping the intermediate My Cart stage. The following source code shows the delivery directly to the Requisition database tables:

LineItem.js

```
366 // EP  
367 // save to DB  
368 // if (!lineItem.saved) mainWnd.requisition.save();  
369 // if (!lineItem.saved) return false;  
370 // if (!mainWnd.requisition.isSaved())  
371 //     return false;  
372 //  
373 // EP  
  
430 // EP  
431 // save to DB  
432 // if (!lineItem.saved) mainWnd.requisition.save();  
433 // if (!lineItem.saved)  
434 //     return false;  
435 // else  
436 return true;
```

437 // EP

punchout.js

```
336 // save to DB  
337 var blnIsSaved = requisition.save()  
338 if (blnIsSaved) {  
339   mainWindow.closeAndSave();  
340 }  
341
```

Requisition.js

```
424 // save to DB  
425 var blnIsSaved = requisition.save()  
426 if (blnIsSaved) {  
427   mainWindow.closeAndSave();  
428 }  
429
```

Because the items must be error free prior to reaching RQ10, Lawson revised the source code to check for errors on an individual product basis upon being selected. This was accomplished by removing the intermediate My Cart save process, and immediately sending items to the database. The removal of the save process is shown below:

```
419 <!-- Print --></button></td>  
420 <!-- EP td><button style="pushbutton" id="saveCartBtn" class="button" onclick="requisition.save()"  
disabled></button></td><!-- Save -->  
421  
422 <td><button style="pushbutton" id=  
423 "checkoutBtn" class="button" onclick=  
424 "requisition.checkout()" disabled>  
425 <!-- Release --></button></td>
```

The “<!--” portion of the code comments out that portion of the code, rendering that portion of the code inoperable.

A comparison of the RQ10.1 fields immediately after an item is selected from the search results shows that items selected in RQC are delivered directly to the requisition, whereas items in RSS are not.

RSS:

Requisition (RQ10.1)

[+] Add [X] Change [Delete] [Previous] [Next] [Inquire]

Requester	RQ1	Bryan Morgan	USD																														
Requisition Number	894	Unreleased	Total																														
<input checked="" type="checkbox"/> Headway <input type="checkbox"/> Miscellaneous <input type="checkbox"/> Line <input type="checkbox"/> Template <input type="checkbox"/> Drop Sheet <input type="checkbox"/> PO Item <input type="checkbox"/> User Fields																																	
<table border="1"> <thead> <tr> <th>Line Item</th> <th>Description</th> <th>Quantity</th> <th>Unit Price</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>1</td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td>2</td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td>3</td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td>4</td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td>5</td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>				Line Item	Description	Quantity	Unit Price	Total	1					2					3					4					5				
Line Item	Description	Quantity	Unit Price	Total																													
1																																	
2																																	
3																																	
4																																	
5																																	

RQC:

Requisition (RQ10.1)

[+] Add [X] Change [Delete] [Previous] [Issue] [Next] [Print]

Requester	RQ1	Bryan Morgan	USD	
Requisition Number	693	Unbilled	Total	12.00
<input checked="" type="checkbox"/> Print Requisition <input type="checkbox"/> Print Lines <input type="checkbox"/> Print Template <input type="checkbox"/> Drop Ship <input type="checkbox"/> PO/BL To <input type="checkbox"/> User Fields				
Item	Description	Quantity	Unit Cost	Yield
1	10001	1	12.00	USD
2		1		USD
3		1		USD
4		1		USD
5		1		USD

The requisition lines mirror the items that are in the requisition. Moreover, the contents of the requisition lines are derived from the requisition. When an item is removed from the requisition shown in the right side of the split screen, the item is also removed for the same requisition shown in RQ10.1, which confirms that the right side of the screen only reflects what is already in the requisition. The removal of the intermediate My Cart step eliminates the means for generating an order list. That removal is significant, more than colorably different from the RSS product, and results in a re-designed product that does not infringe Claim 1 of the '172 patent.

2. Lawson removed the 4th level of UNSPSC code

Lawson restricted the ability of users to search for products by the fourth level of UNSPSC code. This redesign does not infringe claims 3, 28, and 29 of the '683 patent. That modification was implemented by changing the source code in the shoppingrtns.js module.

ePlus based its infringement contentions related to Claims 3, 28, and 29 of the '683 Patent on the fourth level of the UNSPSC code that was available in the RSS system. ePlus claimed that a user's ability to search down to the commodity level met the "means for converting data relating to a selected matching item and an associated source to data relating to an item and a different source" element of Claim 3 of the '683 Patent. ePlus further claimed that searching down to the commodity level met the requirement of Claims 28 and 29 of "converting data relating to a selected matching item and an associated source to data relating to an item and a different source." A screenshot illustrating the commodity level (fourth level) searching from the user's perspective shows that the user can select between "SURGICAL MASK, CLASSICAL" and "SURGICAL MASK, NURSE."

Search for Items by selecting the desired category

Top > Medical Equipment > Medical apparel and > Disposable medicals

- Disposable medical staff isolation or surgical masks
 - Items
- Disposable medical staff shoe covers
 - Items

Add All Selected

= Nonstock item at your From Location = Stock item at your From Location

Add to Cart	Quantity	Description	UOM	Cost	Available
<input type="checkbox"/> SURGICAL MASK, CLASSICAL 458170 1025	<input type="button" value="Add"/> <input checked="" type="checkbox"/> 1 <input type="button" value="Remove"/>	Manufacturer Code: SURGICAL KEND. Number: MASK, CLASSICAL 458170 1025	\$S 1500.00	3,035 EA	
<input type="checkbox"/> SURGICAL MASK, NURSE 458170	<input type="button" value="Add"/> <input checked="" type="checkbox"/> 1 <input type="button" value="Remove"/>	Manufacturer Code: SURGICAL KEND. Number: MASK, NURSE 458170	\$S 36.00	2,800 EA	

RQC was modified from RSS to eliminate the commodity level search, which was the sole basis for ePlus's allegation of infringement. The source code modifications in the shoppingrtns.js module restrict a user's category search to the first six digits of a UNSPSC code and do not allow the user to drill down to the commodity level. Line 133 of the shoppingrtns.js module in RSS (RQC0000750) contains permits drilling down to the value of "5". Line 228 of the shoppingrtns.js module in RQC (RQC0000750), which corresponds to line 133 of the RSS module permits drilling down to the value of "4". The source code related to the modification includes the following:

RSS	RQC
133 if (indexNum > 5)	228 if (indexNum > 4) // only going to the CLASS level.

From a user's perspective, the RQC product does not allow a user to drill down to a level where there are identical or generally equivalent products. As shown by the screenshot from RQC, the user is left with the "SURGICAL MASK" results as well as "SHOE COVERS, ELASTIC TOP." Surgical masks and shoe covers share the same 6 level UNSPSC code, but cannot be considered equivalent, substantially similar, or substitutable.

Action	Quantity	Item	Description	UOM	Cost	Available
Add	1	1034	SURGICAL MASK, CLASSICAL	EA	\$3 1500.00	3,035 EA
Add	1	1035	SURGICAL MASK, NURSE	EA	\$36.00	2,800 EA
Add	1	1022	SHOE COVERS, ELASTIC TOP	EA	\$21.00	-450 EA

The removal of the fourth level (seventh and eighth digits) of the UNSPSC code in the RQC product makes the modification, which is significant, more than colorably different than the RSS product the jury found infringed. ePlus relied exclusively on the eight digit code at trial. That means for converting element and all converting steps have been eliminated.

Lawson incorporates by reference the analysis of the modifications caused by the source code changes as set forth in paragraphs 41-53 of the Declaration of Defendant's Expert Michael I. Shamos, Ph.D, J.D. Concerning Lawson's Non-Infringing Design Around dated September 19, 2011. Further, Lawson incorporates by reference pages 15-19 of Dkt. No. 804, which relates to the removal of the fourth level UNSPSC code.

3. Lawson removed the ability to include items in a single requisition from the Item Master and a Punchout vendor

The RQC product removes the ability of a user to select items from both the Item Master and a Punchout vendor and place them on the same requisition. This modification relates to Configurations 3 and 5 and claims 3, 26, 28, and 29 of the '683 patent. The source code modules that were modified to effect this change included the Cart.js, Dialog.js, inquire.js, Requisition.js and shoppingrtns.js files.

Claim 3 of the '683 Patent requires "at least two product catalogs containing data relating to items associated with the respective sources," a "means for building a requisition using data relating to selected matching items and their associated source(s)" and a "means for processing the requisition to generate one or more purchase orders for the selected matching items." Claims 26 and 28 of the '683 Patent include the steps of "maintaining at least two product catalogs on a database containing data relating to items associated with the respective sources," "building a requisition using data relating to selected matching items and their associated source(s)," and "processing the requisition to generate one or more purchase orders for the selected matching items." Claim 29 of the '683 Patent depends from claim 28. Therefore, if claim 28 is not infringed by any Lawson configuration, neither is claim 29.

ePlus contended at trial that the '683 claims require the ability to generate a requisition containing items from at least two sources. The jury found that Configuration No. 2 (with RSS) did not infringe any of the '683 claims. The sole basis for infringement of the '683 claims was the combination of the RSS product with Punchout as set forth in Configurations 3 and 5. Lawson redesigned the RSS system to prevent the addition of items from both the Item Master and Punchout. The design prevents users who initiate a requisition session with Punchout from adding items from the Item Master as well as users who initiate a requisition session with Item Master from adding items from a Punchout vendor. The source code for the Cart.js, Dialog.js,

inquire.js, Requisition.js and shoppingrtns.js files required modification to implement the change. The below source code from the shoppingrtns.js file (RQC0000750) illustrates the blocking mechanism Lawson implemented to prevent users from placing items from both the Item Master and a Punchout vendor on a single requisition.

```

//-----
111 function doReqSourceTypeChk()
112 {
113 var doNewReq = function(dialogRtn) {
114 if ((dialogRtn.returnValue == "continue") || (dialogRtn.returnValue == "ok")) {
115 mainWnd.progressBar.hide();
116 doCopyHeader();
117 }
118 }
119
120 var msgBreak = (mainWnd.sipObj.styler.showLDS ? "<br/>" : "\n");
121 var msg = "";
122
123 if ((mainWnd.browser.isIE) || (mainWnd.sipObj.styler.showLDS)) {
124 msg = "<center><b><font color='#CD0000'>Punchout items must be on a separate
requisition.</font></b></center>" + msgBreak + msgBreak +
125 "Click <i><b><font color='#1B3F8B'>Cancel</font></b></i> to return to your current requisition" +
126 msgBreak +
127 "Click <i><b><font color='#1B3F8B'>Continue</font></b></i> to create a new requisition";
128 }
129 else{
130 msg = "Punchout items must be on a separate requisition." + msgBreak + msgBreak +
131 "Click Ok to to create a new requisition" +
132 msgBreak +
133 "Click Cancel return to your current requisition";
134 }
135
136 if ((mainWnd.browser.isIE) || (mainWnd.sipObj.styler.showLDS)) {
137 if ("cancel" == mainWnd.dialog.messageBox(msg, "cancelcontinue", "question", null, false, "", doNewReq,
"continue")) {
138 mainWnd.progressBar.hide();
139 return;
140 }
141 else {
142 window.returnValue = "continue";
143 if (mainWnd.browser.isIE)
144 doNewReq.call(this, window);
145 else {
146 if (!mainWnd.sipObj.styler.showLDS)
147 doNewReq.call(this, window);
148 }
149 }
150 }
151 else {

```

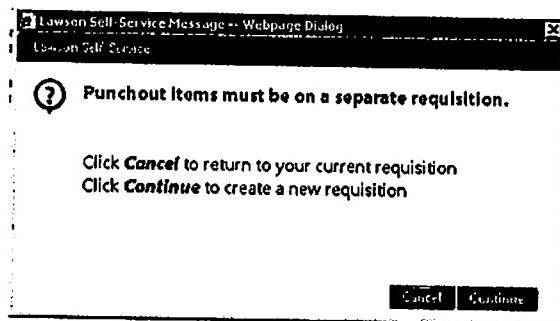
```

152 if ("cancel" == mainWnd.dialog.messageBox(msg, "okcancel", "question", null, false, "", doNewReq, "ok")) {
153 mainWnd.progressBar.hide();
154 return;
155 }
156 else {
157 window.returnValue = "ok";
158 if (mainWnd.browser.isIE)
159 doNewReq.call(this, window);
160 else {
161 if (!mainWnd.sipObj.styler.showLDS)
162 doNewReq.call(this, window);
163 }
164 }
165 }
166 }
167
168 //-----
169 function doCopyHeader()
170 {
171 mainWnd.requisition.checkToDelete();
172 mainWnd.cart.clear(true, false);
173 mainWnd.requisition.isReqCopy = true;
174 mainWnd.InquireInfo.inqCopy = "Y";
175
176 var tkn = "RQIJ.1";
177 var rqQry
178 = "_PDL=" + mainWnd.user.prodline + "&"
179 + "_TKN=" + tkn + "&"
180 + "_EVT=ADD&"
181 + "_RTN=DATA&"
182 + "_TDS=Ignore&"
183 + "_LFN=TRUE&"
184 + "FC=X&"
185 + "RQH-COMPANY=" + mainWnd.requisition.company + "&"
186 + "RQH-REQUESTER=" + mainWnd.user.requester + "&"
187 + "PROCURE-GROUP=" + mainWnd.requisition.procureGroup + "&"
188 + "CPY-HDR-COMM=Y&"
189 + "CPY-HDR-USER-FLDS=Y&"
190 + "ORIG-REQ-NBR=" + mainWnd.requisition.reqNumber + "&"
191 + "ORIG-RQH-COMPANY=" + mainWnd.requisition.company + "&"
192 + "SKIP-LINES=Y&"
193 + "XMLFILE-NAME=" + mainWnd.requisition.cookieId + ".xml";
194
195 mainWnd.callIRQ(X(tkn, rqQry);
196 }
197
198 //-----
199 function displayCategories()
200 {
201 if ((mainWnd.requisition.reqSourceType != RequisitionObject.RQC_SOURCED_REQ) &&
202 (mainWnd.requisition.reqSourceType != "")) {
203 doReqSourceTypeChk();
204 return;

```

205 }
206

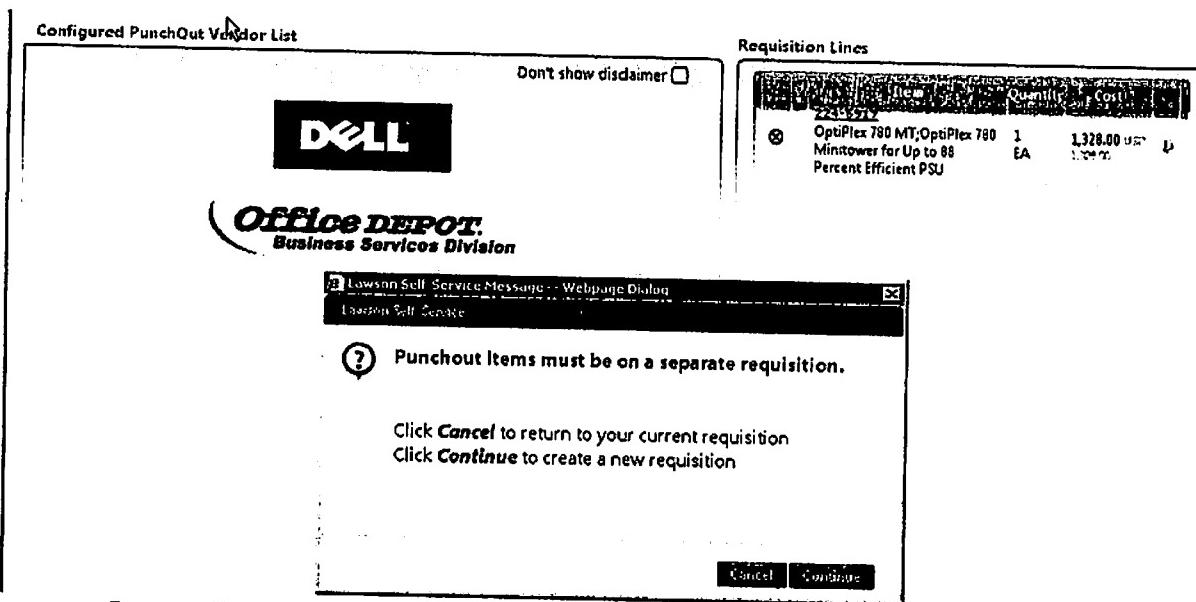
The source code determines the path the user has taken and whether the user has already created a requisition by placing items on that requisition. If that source code determines “yes”, then the user will receive the following warning, which is created if a user already has items from the Item Master and attempts to initiate a Punchout session:



The RQC product is significantly different from RSS because in RQC no requisition can contain items from both Item Master and any Punchout vendor.

The modified RQC source code allows a user options upon receiving the warning message. If the user clears the current requisition such that the new requisition is empty, the system will allow the user to access a Punchout site to search for items. Exhibits F and G of Dr. Shamos’ September 19, 2011, declaration illustrate from the user perspective the source code modifications identified herein. Exhibit F demonstrates the clearing of a prior requisition and creation of a new requisition with no items. Exhibit G demonstrates a Punchout session with a new requisition that contains no items from Item Master.

Exhibit G of the Shamos report further illustrates that a user who selects and places an item on a requisitions from a Punchout vendor cannot thereafter place items on that same requisition from the Item Master. The RQC system will prevent the user from requisitioning a product from the Item Master on the same requisition. The screenshot below shows this significant change:



Lawson incorporates by reference the analysis of the modifications caused by the source code changes as set forth in paragraphs 54-70 of the Declaration of Defendant's Expert Michael I. Shamos, Ph.D, J.D. Concerning Lawson's Non-Infringing Design Around dated September 19, 2011. Further, Lawson incorporates by reference pages 20-25 of Dkt. No. 804, which relates to restricting the placement of items from two Punchout vendors or a Punchout vendor and the item Master.

4. Lawson removed the ability to include items in a single requisition from two Punchout vendors

The RQC product restricts the ability for a user to select items from more than one punchout vendor during a single requisition session. The source code modules that were changes to implement the restriction were CartList.js, Dialog.js, punchout.js,

PunchoutOrderMessage.java, reqCartDetail.js, reqCartDetail.xsl, Requisition.js, RQParser.java, XRQLineDetails.java, and the Lawson 4GL code. Those changes are fully set forth in the source code diff document at RQC0000750. To illustrate, the punchout.js module was modified to restrict any user from placing items on a single requisition from more than one Punchout vendor.

The source code from that module shows the restriction:

```

345 }
346 //-----
347 function processLegalChk()
348 {
349 if (document.getElementById('legalPunch').checked == true)
350 createCookie("RQCrequester_legalPunch", mainWnd.user.requester, 365);
351 else {
352 eraseCookie("RQCrequester_legalPunch");
353 document.getElementById('legalSpan').style.visibility = "hidden";
354 }
355 }
356
357 //-----
358 function doPunchoutGroupIdMsg()
359 {
360 var doNewReq = function(dialogRtn) {
361 if ((dialogRtn.returnValue == "continue") || (dialogRtn.returnValue == "ok")) {
362 mainWnd.progressBar.hide();
363 mainWnd.doCopyHeader();
364 }
365 }
366
367 var msgBreak = (mainWnd.sipObj.styler.showLDS ? "<br/>" : "\n");
368 var msg = "";
369
370 if ((mainWnd.browser.isIE) || (mainWnd.sipObj.styler.showLDS)) {
371 msg = "<center><b><font color='#CD0000'>Each punchout vendor must be on a separate
requisition.</font></b></center>" + msgBreak + msgBreak +
372 "Click <i><b><font color='#1B3F8B'>Cancel</font></b></i> to return to your current requisition" +
373 msgBreak +
374 "Click <i><b><font color='#1B3F8B'>Continue</font></b></i> to create a new requisition";
375 }
376 else{
377 msg = "Each punchout vendor must be on a separate requisition." + msgBreak + msgBreak +
378 "Click Ok to to create a new requisition" +
379 msgBreak +
380 "Click Cancel return to your current requisition";
381 }
382
383 if ((mainWnd.browser.isIE) || (mainWnd.sipObj.styler.showLDS)) {

```

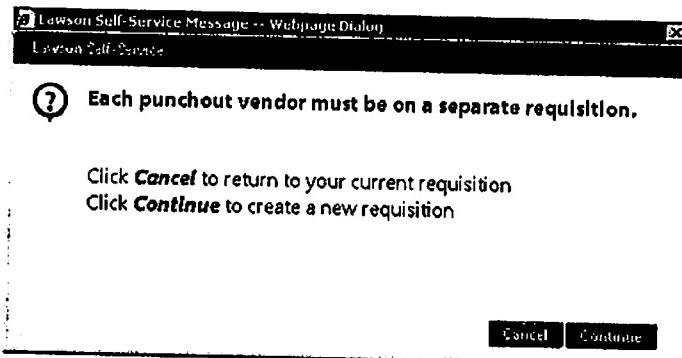
```

384 if ("cancel" == mainWnd.dialog.messageBox(msg, "cancelcontinue", "question", null, false, "", doNewReq,
"continue")) {
385 mainWnd.progressBar.hide();
386 return;
387 }
388 else {
389 window.returnValue = "continue";
390 if (mainWnd.browser.isIE)
391 doNewReq.call(this, window);
392 else {
393 if (!mainWnd.sipObj.styler.showLDS)
394 doNewReq.call(this, window);
395 }
396 }
397 }
398 else {
399 if ("cancel" == mainWnd.dialog.messageBox(msg, "okcancel", "question", null, false, "", doNewReq, "ok")) {
400 mainWnd.progressBar.hide();
401 return;
402 }
403 else {
404 window.returnValue = "ok";
405 if (mainWnd.browser.isIE)
406 doNewReq.call(this, window);
407 else {
408 if (!mainWnd.sipObj.styler.showLDS)
409 doNewReq.call(this, window);
410 }
411 }
412 }
413 }
414
415 //-----
416 function createCookie(name,value,days)
417 {
418 if (days) {
419 var date = new Date();
420 date.setTime(date.getTime()+(days*24*60*60*1000));
421 var expires = "; expires=" + date.toGMTString();
422 }
423 else var expires = "";
424 document.cookie = name + "=" + value + expires + ";" // path=("/");
425 }
426 //-----
427 function readCookie(name)
428 {
429 var nameEQ = name + "=";
430 var ca = document.cookie.split(';');
431 for(var i=0;i < ca.length;i++) {
432 var c = ca[i];
433 while (c.charAt(0)==' ') c = c.substring(1,c.length);
434 if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length,c.length);
435 }

```

```
436 return null;  
437 }  
438  
439 //-----  
440 function eraseCookie(name)  
441 {  
442 createCookie(name,"",-1);  
443 }
```

The system prevents a user from accessing a second Punchout vendor site if there are items from the first Punchout vendor's site within the requisition. Exhibit H from Dr. Shamos' September 19, 2011, declaration shows the restrictions placed on a user who attempts to requisition items from two Punchout vendors on a single requisition. From a user's perspective, the following message will appear: "Each punchout vendor must be on a separate requisition."



The user at this juncture is prevented from visiting another Punchout vendor while any item from a different vendor remains on the requisition. The source code changes in the modules identified in this response make it impossible for RQC with Punchout (Modified Configurations 3 and 5) to generate a requisition containing items from at least two sources from two punchout providers. The modifications to the RSS product to prevent such action makes the RQC product non-infringing and more than colorably different from the RSS product as ePlus argued infringement at trial. That modification is significant.

A full description of the effect of the source code changes are set forth in paragraphs 54-70 of the Declaration of Defendant's Expert Michael I. Shamos, Ph.D, J.D. Concerning Lawson's Non-Infringing Design Around dated September 19, 2011. Further, Lawson incorporates by reference pages 20-25 of Dkt. No. 804, which relates to restricting the placement of items from two Punchout vendors or a Punchout vendor and the item Master.

5. Lawson updated the text depicted on RQC to be consistent with the revisions to the source code

The RQC re-design updated the labeling, visual appearance and messaging to be consistent with the direct to requisition terminology employed by the RQ10 module, which employs functionality the jury found did not infringe. That same direct to requisition technology was brought over to the RQC product. The naming conventions were thus related to that change. The following source code files were modified to bring the RQC program in compliance with the revised label and messaging of that program:

application.xml
approvalListReqs.htm
commonrtns.js
debug.htm
debug.js
debugApprovals.htm
debugRec.htm
debugReqStatus.htm
debugTemplates.htm
default.xml
Dialog.js
ico_cart_green.gif(Deleted)
ico_cart_trans.gif(Deleted)
index.htm
index.htm(2)
indexLocal.htm
inquireLine.htm
inquireOrder.htm
LAWSONLD.shopping.csv
logan.rqc.bkmk(Added)

logan.rqc.bkmk(Deleted)
menus.js
messageDialog.htm
messageDialog.js
messageDialog.js(2)
pdf.htm
PunchoutStepsActionHandler.java
reqLineStatusList.htm
reqStatusList.htm
Requisition.js
rqc_config.xml(Added)
rss_config.xml(Deleted)
RSSShell.js
StylerBase.js
StylerRSS.js
Template.js
templateCopy.htm
templateNew.htm
templatePrint.htm
utility.js
web.xml

These changes were made to reflect the functional changes to the software. For example, because selected items are sent directly to the requisition lines as opposed to a shopping cart, the portion of the source code identifying “My Cart” was changed to “Requisition Lines.”

6. Lawson modified the header related to Punchout sessions

Lawson’s RSS product depicted the customer’s logo at the top left of each screen. When the customer initiated a Punchout session, that branding remained on the screen based on a frameset system. Lawson removed that branding functionality from the RQC product to further show the user that the user was leaving its own website and going to a third party vendor’s website. The source code modules changed were default.xml, punchout.js, punchoutShop.htm, and punchoutShop.js.

To illustrate, the punchoutShop.htm file for RQC contained the following source code:

33 <body onload="punchoutShop_init()" onresize="punchoutShop_resize()"

```
34 onunload="punchoutShop_unload()" style="visibility:hidden">
35 <img id="punchoutLogo" border="0" name="punchoutLogo"><br>
```

That code enabled the frameset to pull the “punchoutLogo” graphic (e.g., a company logo) and place that graphic on the screen when the user punched out to a third party. The RQC product removed that source code as identified in the punchoutShop.htm module in the document RQC0000750. The other source code files had similar modifications which are set forth in those modules.

7. Lawson provided an information screen to inform the customers that they were leaving their system and going to a third party vendor system

The RQC software employs a warning message to the user that the user is leaving the customer’s website and going to a third party website. The source code changes related to that modification are set forth in the modules punchout.htm and punchout.js. The punchout.js source code (RQC0000750) that provides the notification is set forth below:

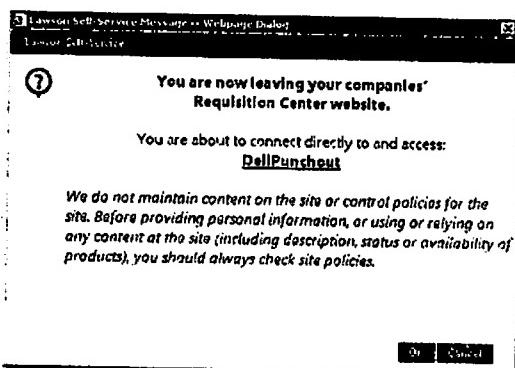
```
201 mainWnd.progressBar.hide();
202 }
203
204 var msgBreak = (mainWnd.sipObj.styler.showLDS ? "<br/>" : "\n");
205 var leaveMsg = "<b>You are now leaving your companies'" + msgBreak + "Requisition Center website.</b>" +
msgBreak;
206 var accessMsg = "You are about to connect directly to and access:" + msgBreak + "<b><u>" +
mainWnd.cmnGetNodeCDataValue(recordNodes[idx].getElementsByTagName("COL")[2]) + "</u></b>" +
msgBreak + msgBreak;
207 var disclaimerMsg = "<i>We do not maintain content on the site or control policies for the site. " +
208 + "Before providing personal information, or using or relying on any content at the site " +
209 + "(including description, status or availability of products), you should always check site policies.</i>";
210 var msg = "";
211 if ((mainWnd.browser.isIE) || (mainWnd.sipObj.styler.showLDS))
212 msg = "<center>" + leaveMsg + msgBreak + accessMsg + "</center>" + "<font color='#C67171'>" +
disclaimerMsg + "</font>";
213 else {
214 // For Firefox
215 msg = "You are now leaving your companies' Requisition Center website." + msgBreak + msgBreak
216 + "You are about to connect directly to and access:" + msgBreak +
mainWnd.cmnGetNodeCDataValue(recordNodes[idx].getElementsByTagName("COL")[2]) + msgBreak + msgBreak
217 + "We do not maintain content on the site or control policies for the site. " +
218 + "Before providing personal information, or using or relying on any content at the site " +
219 + "(including description, status or availability of products), you should always check site policies.";
220 }
```

```

221
222 xCookie = readCookie('RQCrequester_legalPunch');
223 if (!xCookie) {
224 if ("cancel" == mainWnd.dialog.messageBox(msg, "okcancel", "question", null, false, "", processVendorClick)) {
225 mainWnd.progressBar.hide();
226 return;
227 }
228 else {
229 document.getElementById('legalSpan').style.visibility = "visible";
230 window.returnValue = "ok";
231 if(mainWnd.browser.isIE)
232 processVendorClick.call(this, window);
233 else {
234 if (!mainWnd.sipObj.styler.showLDS)
235 processVendorClick.call(this, window);
236 }
237 }
238 )
239 else {
240 window.returnValue = "ok";
241 processVendorClick.call(this, window);
242 }
243 }
244
245

```

As with all of the source code changes identified in response to Interrogatory No. 5, they must be read in context with the source code in the particular modules as well as the interaction with other source code modules. From a user's perspective, a screen message appears informing the user that it is leaving the company website.



The message set forth above is delivered to each unique user. A particular user may elect not to see the message during later Punchout sessions. A single user's election does not affect any other user.

AS TO OBJECTIONS

Dated: October 24, 2011

LAWSON SOFTWARE, INC.

By _____/s/

Of Counsel

Dabney J. Carr, IV (VSB No. 28679)
Robert A. Angle (VSB No. 37691)
Megan C. Rahman (VSB No. 42678)
dabney.carr@troutmansanders.com
robert.angle@troutmansanders.com
megan.rahman@troutmansanders.com
TROUTMAN SANDERS LLP
1001 Haxall Point, Richmond, VA 23219
Telephone: (804) 697-1200
Facsimile: (804) 697-1339

Daniel McDonald (admitted *pro hac vice*)
William D. Schultz (admitted *pro hac vice*)
Rachel C. Hughey (admitted *pro hac vice*)
Andrew J. Lagatta (admitted *pro hac vice*)
MERCHANT & GOULD P.C.
3200 IDS Center, 80 South Eighth Street,
Minneapolis, MN 55402
Telephone: (612) 332-5300
Facsimile: (612) 332-9081

Donald R. Dunner (admitted *pro hac vice*)
Erika H. Arner (admitted *pro hac vice*)
**FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.**
901 New York Avenue, N.W.
Washington, DC 20001

Telephone: (202) 408-4000
Facsimile: (202) 408-4400

Counsel for Defendant Lawson Software, Inc.